

**R E M A R K S**

This amendment is responsive to the Office Action mailed on June 7, 2004. The specification was objected to because the disclosure contained an embedded hyperlink; Applicants amended the specification to remove the hyperlink. The specification was further objected to because of informalities: the missing quotation marks on page 4, line 22 have been added by way of amending the specification; however, the exclamation point has not been removed and replaced with a period since an exclamation point should be acceptable, proper punctuation to the same extent that commas, colons, and semicolons are acceptable and proper. Applicants wished to punctuate the comment with which the exclamation point was used. If the Examiner can offer an explanation regarding why this punctuation should be changed, Applicants shall be pleased to reconsider this request.

Applicants are confused by the Examiner's request for formal drawings since formal drawings were filed with the specification and claims on March 26, 2001. Applicant submits herewith another set of formal drawings for the Examiner's convenience or to replace the previously-filed drawings should there be an informality about them of which Applicants are unaware.

The Examiner requests that Figs. 2A and 2B "be designated by a legend such as -- Prior Art—because only that which is old is illustrated" (office action page 3). Applicants respectfully decline to make this change at this time for the following reasons. First of all, there already are legends in these Figs. Fig 2A is clearly designated "LEGACY ARCHITECTURE EXAMPLE" and Fig. 2B is clearly designated

“CIM/XML ARCHITECTURE”. Applicants believe that these are appropriate and descriptive legends for these Figs., although these Figs. are discussed in the Background section of the specification. Furthermore, although these software architectures existed when Applicants made the subject matter recited in the instant claims, these are merely representations of legacy and new-standard software architectures. While Applicants’ claims do recite software architecture, the claims recite subject matter well beyond mere software architectures. Thus, to label Figs. 2A/2B “Prior Art” may be misleading to a reader of any patent issuing from the instant application because that could imply that the claims are directed to new and improved software architecture, *per se*, which is not the case. In Applicants’ opinion, the public shall be better served to have these Figs. designated as is. The Examiner is respectfully requested to reconsider his position on this point. Accordingly, although Applicants adopt herein the Examiner’s phraseology “admitted prior art” in reference to both Figs. 2A/2B and schema included therein for purposes of facilitating communication on the record, Applicants do not necessarily agree that Figs. 2A/B are prior art vis-à-vis the claimed subject matter.

Claims 1-89 were presented for examination and were rejected. No claims are canceled and no claims are added. Claims 1-89 remain pending. Claims 1, 12, 24, 37, 49, 52, 59, 63, 68, 70, 75 and 81 are independent claims; all have been amended to be limited to software architecture, and some have been further amended to improve form. Dependent claims 27, 34, 36, 39, 45, 46, and 48 have been amended to improve form.

Claims 11, 23, 37, 51, 52, 74, and 88 were rejected under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention. Each of these claims has been

amended in accordance with the Examiner's commentary in section 8 of the Office Action and in a manner intended to overcome this rejection.

35 U.S.C. § 103

Claims 1-62 and 75-89 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,086,622 to Abe et al., (hereinafter "Abe"), in view of admitted prior art and further in view of U.S. Patent No. 5,295,256 to Bapat (hereinafter "Bapat"). Claims 63-74 are rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,086,622 to Abe et al., (hereinafter "Abe"), in view of admitted prior art. Applicants respectfully traverse these rejections for the following reasons.

Independent Claims 1, 12, 24, 37, 49, 52, 59, and 63

To begin with the purposes of Abe and Applicants' claimed subject matter are very different. A principal purpose of Applicant is to provide a translator-compiler for converting legacy management software compatible with legacy or proprietary architecture (software architecture such as that shown in Fig. 2A) to compatibility with preferred, standard, non-legacy architecture (software architecture such as that shown in Fig. 2B). But Abe, by contrast, teaches compiling a high level language operable with a first hardware architecture into machine code operable with a second and non-existent hardware architecture, along with some other steps of decompiling and linking for purposes of allowing debugging of the machine code prior to the availability of the second hardware architecture for which it is actually intended. The purposes of Applicant and Abe are thus headed in wildly different directions.

In accordance with these different purposes, it is clear that Abe does not disclose or suggest a system employing "*management software*" compatible with first architecture

and not compatible with second architecture. Abe discloses a system employing “*debugging software*” compatible with first architecture and (arguably) not compatible with second architecture, and it is well known in the computer industry that these two software products are in entirely different categories. Management software is used by a computer system that is currently operational and functional; it can be used, for example, for keeping track of various functional operations involving storage, printers, servers, etc. within the computer system, as it is then running. By stark contrast, debugging software is used BEFORE a system becomes operational; it is used to find the problems in a system that is under development or which has just been developed but which is not yet fully functional and without a history of successful operation. The two categories of software are entirely different. Each of the independent claims 1, 12, 24, 37, 49, 52, 59, and 63 recites *management software*. Abe does not disclose or suggest management software within the context in which management software is recited in any of these claims. Furthermore, neither “admitted prior art” or Bapat cures this deficiency of Abe. Accordingly, the rejection under 35 U.S.C. § 103(a) of these claims should be withdrawn on the basis of this argument alone. Moreover, claim sets 2-11, 13-23, 25-36, 38-48, 50-51, 53-58, 60-62, and 64-67 are dependent claims which depend, respectively, from these independent claims and the rejection of these dependent claims should also be withdrawn, at least for reasons based on their dependency. Accordingly, it is respectfully submitted that claims 1-67 are allowable.

All Independent Claims Including The Foregoing:

Abe is hardware-oriented and is directed to the debugging of machine code to be used on new hardware architecture prior to availability of the new hardware architecture.

“When a *computer of a new architecture* which is different from a present architecture is developed, it is necessary to debug a *machine program* for the computer. However, in general, when the machine program for the new computer is to be debugged, a *new computer (hardware)* is not completely produced, i.e., does not actually exist. Therefore, it is impossible to actually debug the machine program on the new computer.” (Abe, column 1, lines 17-24, emphasis added)

Thus, the problem to which Abe is directed has to do with hardware that will not be ready until a future date. Abe wants to be able to debug a *machine program* intended for the non-existing, new hardware architecture on presently-existing, similar hardware architecture. A machine program is machine code or “1’s” and “0’s” and is the code which the hardware understands and which makes it operate.

“In the following descriptions, a computer of an existing architecture (hereinafter referred to as a first architecture) is a *serial executing type* and a computer of a new architecture (hereinafter referred to as a second architecture) is a *parallel executing type*.” (Abe, column 4, lines 34-39, emphasis added)

Thus, the example relied upon throughout Abe is converting from a serial hardware architecture to a parallel hardware architecture, where the serial machine is in existence but the new, parallel machine does not yet exist. Abe has a high level language program which it compiles for the new, parallel architecture; Abe decompiles it and reassembles it to run on the currently-available, serial architecture to test the new machine code on the existing serial architecture, rather than wait for the parallel architecture to be ready.

Thus, in view of the above, when Abe refers to converting a program for a computer of a first architecture to a program for a computer of a second architecture, he is referring to a first hardware architecture and a second hardware architecture. As further evidence of the limitation of the disclosure in Abe to hardware architecture, consider:

“a first step of compiling a first high-level language source program for a computer of a first architecture, *thereby producing a machine program for a computer of a second architecture*” (Abe, column 2, lines 5-8, and independent claims 1 and 10, emphasis added).

In other words, Abe discloses and recites in every claim, a step whereby he compiles a high level language. This is a high level language for a computer having a first architecture. This compiling step produces a machine program or machine code. But it produces machine code for a computer having a *different* architecture. The only kind of architecture that could possibly fit this description is *hardware* architecture because one does not and cannot produce machine code (“1’s” and “0’s”) for software architecture. Software architecture doesn’t run machine code; only hardware architecture runs machine code. Therefore one could not produce machine code for a second architecture based merely on compiling a high level language for a first architecture *unless* the second architecture is *also* hardware architecture. It is thus clear that hardware architecture exclusively, and not software architecture, is being referred-to in Abe.

By contrast, Applicants’ disclosure and claims are limited to *software*. The title is: “Translator-Compiler for Converting Legacy Management *Software*” (emphasis added). Software is discussed throughout the application: In reference to “legacy or proprietary architecture” Applicants’ disclosure says: “These architectures are combinations of *software*, such as schemas, languages, and protocols, etc.” (application, page 4, lines 2-3). In Figs. 2A and 2B, legacy *architecture* and CIM/XML *architecture* are depicted and these are the kinds of architectures to which Applicants’ claims relate. Clearly, these are architecture stacks of software; no hardware is depicted in Figs. 2A/2B.

Consider Applicants' amended claims. Each and every independent claim has been amended to recite "software architecture". For example, claim 1 recites, *inter-alia*: "A computer system employing management software written in a first computer language compatible with first software architecture and not compatible with second software architecture.....". This limitation clearly defines around the disclosure of Abe, the primary or teaching reference, which is limited to hardware architecture as demonstrated above. MPEP 2143 says that to establish a *prima facie* case of obviousness, three basic criteria must *all* be met. First, there must be some suggestion or motivation, either in the reference themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings. And, second, there must be a reasonable expectation of success. And, third, the prior art reference or references when combined must teach or suggest all the claim limitations. It is clear that Abe does not teach or suggest all claim limitations because it was not applied against all of the claim limitations in the first place, and it does not teach or suggest software architecture, as recited in claim 1. The "admitted prior art" and Bapat, taken with Abe in any reasonable combination, fail to cure the deficiencies in Abe, and are not combinable with Abe in any event for reasons to be explained below.

The requisite motivation to combine the "admitted prior art" and Abe is lacking. Page 5 of the Office Action indicates that "Abe does not explicitly disclose a schema formed within the first architecture" or "header files contained within the schema". Applicant agrees that Abe is deficient in disclosing or suggesting a schema or header files. And that isn't surprising in view of Abe's discussion, which is limited to hardware-oriented architectures used in the debugging of machine code. Applicant's discussion of

software schemas are part of the legacy software architecture discussed in Applicants' "admitted prior art" (within the background section of the application). It was the desire to make use of this legacy (old software architecture) software in a readily facile manner that inspired the solution described by Applicants' claimed subject matter. Applicants solution has absolutely nothing to do with generating and debugging machine code intended for hardware architecture yet to be manifested. This attempted combination goes beyond any suggestion that can be gleaned from Abe, which doesn't need or hint at schemas to accomplish its objective, or gleaned from the "admitted prior art" which doesn't suggest usage of schemas in any manner other than as part of legacy software. Thus, there is no motivation to combine these references to be derived from the references themselves. Any motivation would come from improper hindsight reasoning.

Moreover, there is no reasonable expectation of success for the combination of admitted prior art and Abe. The Office Action suggests to combine incompatible quantities. As noted, there is no suggestion in Abe that a schema is needed or wanted. Abe discusses a compiler used to generate machine code, a decompiler to generate high level code from the machine code, and compiling and linking the second high level language source program to produce a first executable load module. This *hardware environment* of Abe does not suggest that a *schema (header-related software)* could be successfully or usefully integrated. The Office Action has reached into the software realm of legacy software to come up with the notion of a schema formed within software architecture, to imagine a combination of that schema with the hardware architecture of Abe. The Office Action thus reflects improper hindsight reasoning, ignores the clear lack

of motivation in either reference to make this combination noted above, and further ignores the improbability of success of making this combination.

The Office Action, page 6, indicates that neither Abe nor Applicant's "admitted prior art" explicitly discloses manipulating header files to locate public functions and/or data attributes of the header files, or emitting code that calls the public functions and/or data attributes in the first language to obtain called public function and/or data attributes. Applicants agree with the Examiner, that both Abe and Applicants' "admitted prior art" are deficient in disclosing or suggesting manipulating header files, or emitting code, for these purposes. And that isn't surprising in view of Abe's discussion which is limited to hardware-oriented architectures used in the debugging of machine code.

The Office Action then applies Bapat and says it discloses an analogous computer system where it would have been obvious to incorporate the method of manipulating header files as taught by Bapat into the method of converting a program of a first architecture to a second architecture as taught by the combination system of Abe and "admitted prior art", etc. However, Bapat is NOT a compatible disclosure with Abe. Bapat is based on object oriented programming (OOP) and specifically calls out the C++ language. *See* Bapat, for example, column 5, lines 51, 63, 68 and other places. But the high level language in Abe is "C" which is NOT an object oriented language. *See* Abe, for example, column 4, line 43; column 6, lines 32, 37, 46 and other places where it refers to C; it also refers to FORTRAN and COBOL in column 7, lines 10-17, which also are not object oriented languages. Although C++ is a superset of C, where use of a C++ compiler can make use of C language source code, the reverse is not true. C++ (the language in Bapat) cannot be directly converted into equivalent C language (the language

in Abe). Thus, one of ordinary skill in the art reading Bapat would conclude that Bapat is irrelevant in this regard. In view of the above there is no motivation to combine these references to be derived from the references themselves. And, any header files in Bapat would be in C++ which could not be brought into Abe, offering *no* expectation of success.

The determination of *prima facie* obviousness must be supported by a finding of “substantial evidence.” *See In re Zurko*, 258 F.3d 1379, 1386 (Fed. Cir. 2001). Specifically, unless “substantial evidence” found in the record supports the factual determinations central to the issue of patentability, including motivation and expectation of success, the rejection is improper and should be withdrawn. In view of the above, Applicants submit that that there is no “substantial evidence” in the record to support the factual determinations alleged by the Examiner with respect to motivation and/or expectation of success in combining “admitted prior art” with Abe, or in combining Bapat with both “admitted prior art” and Abe.

Accordingly, the rejection of claim 1 under 35 U.S.C. § 103(a) as being unpatentable over Abe in view of admitted prior art and further in view of Bapat should be withdrawn and the claim allowed.

All of the other independent claims in the application, namely: 12, 24, 37, 49, 52, 59, 63, 68, 70, 75, and 81 are rejected under 35 U.S.C. § 103(a) as being unpatentable over either Abe in view of “admitted prior art” in further view of Bapat, or over Abe in view of “admitted prior art”. All of the other independent claims recite subject matter similar or related to that of claim 1 and they have all been amended by changing “architecture” to “software architecture” in a manner similar to that of claim 1.

Therefore, the rejection of these independent claims should be withdrawn for reasons similar to those given above, and the claims should be allowed.

Since each set of dependent claims, namely: claim sets 2-11, 13-23, 25-36, 38-48, 50-51, 53-58, 60-62, 64-67, 69, 71-74, 76-80, and 82-89 depends from its respective independent claim which is believed to be allowable, the dependent claims should be allowable also, at least for reasons of their dependent status. These dependent claims are independently patentable because of their individual claim language recitations, but discussion of the reasons therefor is moot in light of the above.

.....

## **CONCLUSION**

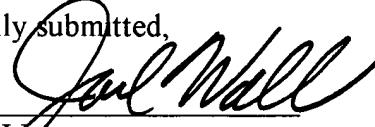
In view of the above arguments, the rejection of claims 1-89 under 35 U.S.C §103(a) should be withdrawn. To the extent that the above-discussed or any other Office Action citations of Abe, “admitted prior art”, and/or Bapat were applied against particular independent and dependent claim elements but not expressly rebutted herein, it is to be understood that Applicants’ silence does not mean or imply acquiescence. Rather, Applicants believe that any such response to application of such citations would be moot in view of the foregoing arguments and provisions of MPEP § 2143.

Applicants agree with the Examiner’s decision to not rely upon any of the prior art of record in rejection of Applicants’ claims since, in Applicants’ view, those references taken alone, or in any reasonable combination, do not disclose or suggest

Applicants' claims. Reconsideration and allowance of claims 1-89 are therefore respectfully requested.

To the extent that an extension of time may be needed in order to enter this amendment in this case, please consider this response as including a petition under 37 C.F.R. § 1.136 for such extension of time. Please charge any fee for such petition or any other fee or cost that may be incurred by way of this amendment to Patent Office deposit account number 05-0889. If the Examiner feels that a telephone conversation may serve to advance the prosecution of this application, he or she is invited to telephone Applicants' undersigned representative at the telephone number provided below.

Respectfully submitted,

  
JOEL WALL  
Registration No. 25,648

TELEPHONE:  
JOEL WALL ESQ.  
508-625-1323  
August 2, 2004